

In-situ multi-scattering tomography

Vadim Holodovsky,^{1*} Yoav Y. Schechner,¹ Anat Levin,²
Aviad Levis,¹ Amit Aides,¹

¹Dept of Electrical Engineering,
Technion - Israel Institute of Technology,
Haifa 32000, Israel

²Dept of Mathematics and Computer Science,
Weizmann - Institute of Science,
Rehovot, 76100 Israel
{vholod@tx.technion.ac.il}

December 2, 2015

Abstract

To recover the three dimensional (3D) volumetric distribution of matter in an object, images of the object are captured from multiple directions and locations. Using these images tomographic computations extract the distribution. In highly scattering media and constrained, natural irradiance, tomography must explicitly account for off-axis scattering. Furthermore, the tomographic model and recovery must function when imaging is done in-situ, as occurs in medical imaging and ground-based atmospheric sensing. We formulate tomography that handles arbitrary orders of scattering, using a monte-carlo model. Moreover, the model is highly parallelizable in our formulation. This enables large scale rendering and recovery of volumetric scenes having a large number of variables. We solve stability and conditioning problems that stem from radiative transfer (RT) modeling in-situ.

1 Introduction

Recovering scenes via participating media [1–6] often focus on observing background objects [7–9]. However, it is also important to recover the medium itself, as done in remote sensing of the atmosphere. Recent works seek volumetric recovery of a three dimensional (3D) heterogeneous scattering media, focusing on the atmosphere. Being very large, recovery of the atmosphere generally requires passive imaging, using the steady, uniform and collimated Sun as the radiation source. The data is images acquired from multiple directions [10], which sample the scene’s light-field.

Based on multi-view image data, computational tomography (CT) yields 3D volumetric recovery in many domains [11–14], including biomedical imaging. However, in most CT models, as in X-ray, direct-transmission [15] forms the signal, while small-angle scattering has been considered to be a perturbation. In contrast, in a medium as the atmosphere, the source (unidirectional sun) and detector (wide angle camera) are generally not aligned: *scattering*

including high orders is the signal.¹ For this reason, there is a need to formulate tomography based explicitly on multiple scattering. This work advances this formulation.

Ref. [16] assumes that the signal is mostly a result of single-scattering, thus deriving single-scattering tomography. However in many real situations high order scattering is not negligible. Our current work generalizes [16] to multiple scattering. On the other hand, Refs. [17,18] focus on a diffusion limit, equivalent to infinite scattering orders. However, many scenes of interest do not comply with the diffusion limit, having regions of low-order scattering. The tomography work we now describe handles any order of scattering events, due to use of a monte-carlo (MC) model. The approach of [19] performs tomography, where radiative transfer is based on a discrete ordinate spherical harmonic method. While MC can realize scattering events in any location and direction, discrete ordinate methods are by definition constrained to discretized or band-limited propagation. The medium in [19] is captured from afar: there are no scattering events near the camera.

This paper derives multi-scattering 3D tomography, where cameras can be arbitrarily close to the medium, and in fact can be in-situ, as the system in [10]. As we explain, such a setup imposes instabilities on the image formation forward model, which can strongly affect recovery. Moreover, a forward model involving multiple scattering is computationally complex. Attempting an inverse-problem generally magnifies computational complexity, jeopardizing its realistic prospects. This work addresses all these issues. First, we propose a way to stabilize the forward model, including in-situ image rendering, while being computationally efficient. Efficiency is achieved using a forward-MC principle, which parallelizes calculations to multi-view cameras, for each photon packet. This gain is in addition to the inherent parallelizable nature of MC, where each photon packet can traverse the medium independent of other packets. Second, in the inverse problem, we use efficient optimization using surrogate functions [19], while solving an ill-conditioned formulation that arises from the in-situ setup.

2 Theoretical background

In this section we describe the basic building blocks of radiative transfer through a non-emitting medium. Using these building blocks, two common Monte-Carlo (MC) methods are described, each having a specific advantage and disadvantage, complementing to the other method, in the context of multiview in-situ imaging. Consequently, in Sec. 3, we derive a new MC method which better addresses the setup.

Extinction: Radiance is a flow of photons. Light propagation through the atmosphere is affected by interaction with air molecules and aerosols (airborne particles). Atmospheric constituents have an *extinction cross section* for interaction with each individual photon. Per unit volume, the *extinction coefficient* due to aerosols is $\beta^{\text{aerosol}} = \sigma^{\text{aerosol}} n$. Here σ^{aerosol} denotes aerosol extinction cross section and n denotes particle density. The total extinction is a sum of the aerosol and molecular contributions, $\beta = \beta^{\text{aerosol}} + \beta^{\text{air}}$, where β^{air} is modeled as a function of altitude and wavelength λ [16]. The *optical depth* along a photon path S is

$$\tau = \int_S d\tau = \int_S (\beta^{\text{aerosol}} + \beta^{\text{air}}) dl = \int_S (\sigma^{\text{aerosol}} n + \beta^{\text{air}}) dl = \tau^{\text{air}} + \int_S \sigma^{\text{aerosol}} n dl, \quad (1)$$

where $\tau^{\text{air}} = \int \beta^{\text{air}} dl$. The fraction of radiation power that gets transmitted through the atmosphere is the *transmittance* t , which exponentially decays with the optical depth (Beer-Lambert law):

$$t = \exp(-\tau). \quad (2)$$

¹In a *single scattering* regime, each light ray changes direction at most once due to scattering. In a multiple scattering regime, light can change direction by scattering in multiple events (orders).

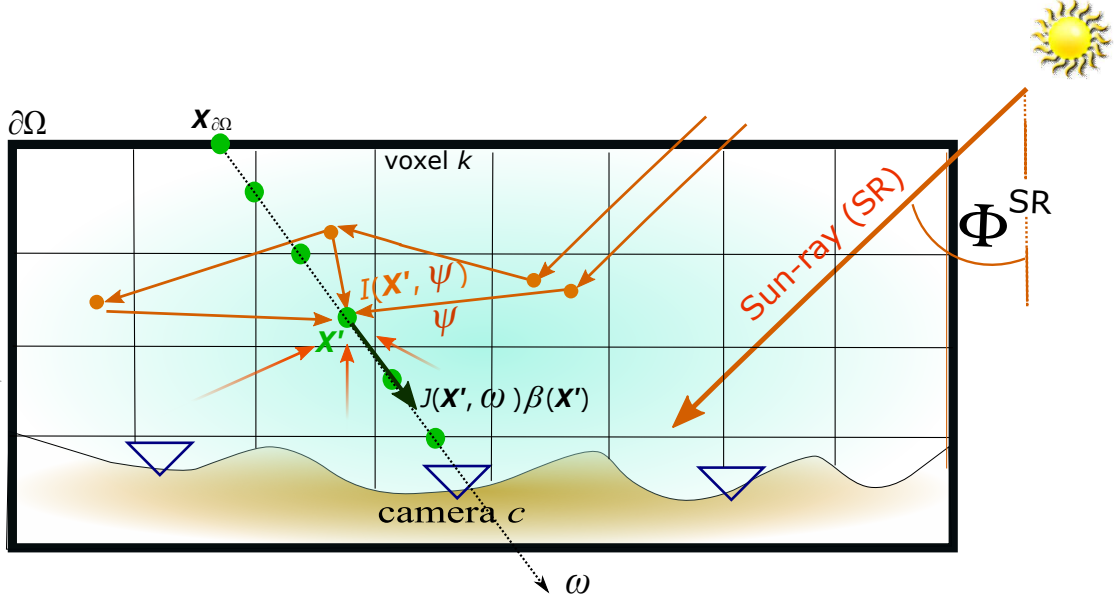


Figure 1: Integral (lightfield) imaging through a volumetric distribution in the atmosphere, using ground-based cameras.

Scattering: Suppose that a photon interacts with a single particle. The unitless *single scattering albedo* ϖ of the particle, determines a probability for scattering. The aerosol single scattering albedo is ϖ^{aerosol} . The *scattering coefficient* due to aerosols in the volume is $\alpha^{\text{aerosol}} = \varpi^{\text{aerosol}} \beta^{\text{aerosol}} = \varpi^{\text{aerosol}} \sigma^{\text{aerosol}} n$. For non-isotropic scattering, an angular function defines the probability of photons to scatter into each direction. Let $\omega, \psi \in \mathbb{S}^2$ (unit sphere) represent photon or ray directions. The fraction of energy scattered from direction ψ towards direction ω is determined by a *phase function* $P(\omega \cdot \psi)$. The phase function is normalized: its integral over all solid angles is unity, and is often approximated by a parametric expression. Specifically, the Henyey-Greenstein function, parameterized by an *anisotropy parameter* $-1 \leq g \leq 1$, can approximate aerosol scattering [16]

$$P_{\text{HG}}(\omega \cdot \psi) = \frac{1}{4\pi} \frac{1 - g^2}{[1 + g^2 - 2g(\omega \cdot \psi)]^{\frac{3}{2}}}. \quad (3)$$

Scattering by air molecules follows the *Rayleigh* model

$$P_{\text{Ray}}(\omega \cdot \psi) = \frac{3}{16\pi} [1 + (\omega \cdot \psi)^2]. \quad (4)$$

In the visible range, air single scattering albedo is $\varpi^{\text{air}} \simeq 1$ and emission is negligible. For simplicity, wavelength dependency is omitted.

Radiative Transfer Equation: Denote the volumetric extinction field at position \mathbf{X} by $\beta(\mathbf{X})$. The radiative transfer equation (RTE) describes the flow of radiance $I(\mathbf{X}, \omega)$ at \mathbf{X} , through the medium [20]

$$\nabla_{\omega} I(\mathbf{X}, \omega) = -\beta(\mathbf{X}) I(\mathbf{X}, \omega) + \beta(\mathbf{X}) J(\mathbf{X}, \omega). \quad (5)$$

Here $J(\mathbf{X}, \boldsymbol{\omega})$ is the *in scattering* [20] volumetric field

$$J(\mathbf{X}, \boldsymbol{\omega}) = \varpi \int_{4\pi} P(\boldsymbol{\omega}, \boldsymbol{\psi}) I(\mathbf{X}, \boldsymbol{\psi}) d\boldsymbol{\psi}. \quad (6)$$

Denote the medium boundary by $\partial\Omega$ and the boundary radiance as $I_{\partial\Omega}$. Let $\mathbf{X}_{\partial\Omega}$ be the intersection point of the boundary with a ray in direction $\boldsymbol{\omega}$. Integrating Eq. (5) along direction $\boldsymbol{\omega}$ defines the integral form of the RTE (Fig. 1)

$$I(\mathbf{X}, \boldsymbol{\omega}) = I_{\partial\Omega} \exp\left[-\int_{\mathbf{X}}^{\mathbf{X}_{\partial\Omega}} \beta(r) dr\right] + \int_{\mathbf{X}}^{\mathbf{X}_{\partial\Omega}} J(\mathbf{X}', \boldsymbol{\omega}) \beta(\mathbf{X}') \exp\left[-\int_{\mathbf{X}}^{\mathbf{X}'} \beta(r) dr\right] d\mathbf{X}'. \quad (7)$$

MC is a popular numerical approach to solve Eqs. (5,6,7).

2.1 Monte Carlo Photon Tracking

MC methods trace propagated photons, given the source radiance and $\varpi, P(\boldsymbol{\omega}, \boldsymbol{\psi}), \beta$. The propagation realizes Eq. (7). The result is an estimate of the radiance around $(\mathbf{X}, \boldsymbol{\omega})$, across the domain. In our case study, the light source (sun) is effectively located at infinity and light is captured by cameras. A modeled camera c has center of projection at location \mathbf{X}_c . Each pixel p collects radiation flowing from a narrow cone around direction $\boldsymbol{\omega}_p$, yielding a raw image $i_c(\boldsymbol{\omega}_p)$. In order to derive images $i_c(\boldsymbol{\omega}_p)$, we describe two existing MC approaches [22].

1. Forward Monte Carlo (FMC): photons propagate from the source (sun) to the detector.
2. Backward Monte Carlo (BMC): photons propagate from the detector to the source.

2.1.1 Sampling by Inverse Transform

MC is stochastic. It treats scattering and extinction as random phenomena sampled from probability distributions. Random sampling at the heart of MC is realized by an *inverse transform* [21] of a specified probability density function. We now briefly describe this mathematical process. Let u be a random number drawn from a uniform distribution in the unit interval: $u \sim \mathcal{U}[0, 1]$. The number u can be transformed into a random variable χ , whose cumulative distribution function (CDF) is $F(\chi)$. The transform is defined by $\chi = F^{-1}(u)$, where F^{-1} denotes the inverse of F . Specifically, consider a photon propagating in the atmosphere. The photon has high probability of propagating as long as t is high, but the probability diminishes as $t \rightarrow 0$. Thus Eq. (2) can be viewed as a probability density function, whose CDF is

$$F(\tau) = \int_0^\tau \exp(-\tau') d\tau' = 1 - \exp(-\tau). \quad (8)$$

Each photon then propagates to a random optical depth

$$\tau^{\text{random}} = F^{-1}(u) = -\ln(1 - u). \quad (9)$$

2.1.2 Forward Monte Carlo Photon Tracking

In this approach, photons are generated at the source, illuminating the top of the atmosphere (TOA) uniformly. Photons propagate from the TOA in direction $\boldsymbol{\omega}_{\text{sun}}$. Each photon is traced through the atmosphere. Photons that happen to reach camera c about direction $\boldsymbol{\omega}_p$ are counted as a contribution to $i_c(\boldsymbol{\omega}_p)$. A photon's life cycle is then defined by the following steps (Fig. 2):

(i) Launch a photon-packet from the TOA in direction $\boldsymbol{\omega}_{\text{sun}}$. This is the initial *ray*, denoted \mathcal{R}_0 . The packet has initial intensity I_0 .

Per iteration s :

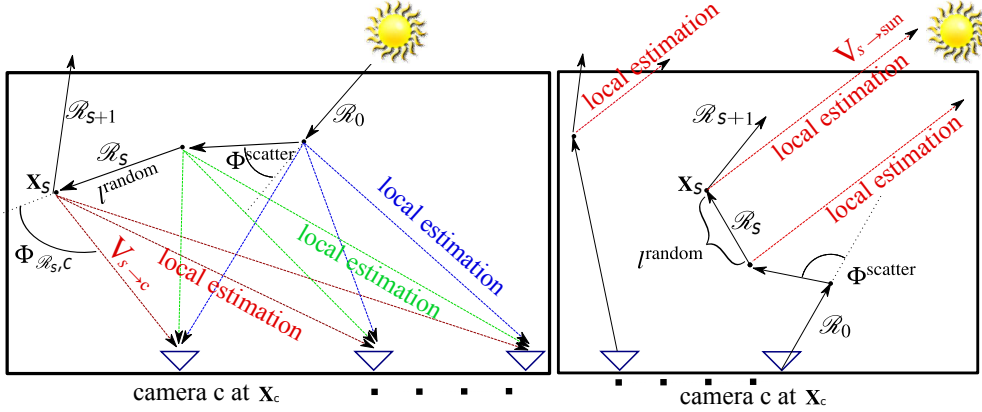


Figure 2: [Left] Multi-view FMC with local estimation. [Right] Multi-view BMC with local estimation..

(ii) Find the distance on ray \mathcal{R}_s to which the photon-packet propagates. Eq. (9) yields τ^{random} . Then using Eq. (1), numerically seek l^{random} s.t.

$$\int_0^{l^{\text{random}}} (\sigma^{\text{aerosol}} n + \beta^{\text{air}}) dl = \tau^{\text{random}}. \quad (10)$$

Distance l^{random} along \mathcal{R}_s yields the 3D position \mathbf{X}_s .

(iii) If \mathbf{X}_s is outside the domain, the packet is terminated. If \mathcal{R}_s passes through \mathbf{X}_c , or a small area around \mathbf{X}_c , the packet is counted as contributing to the image pixel.

(iv) Suppose \mathbf{X}_s is inside the domain. The type of particle (air molecule or aerosol) that the photon-packet interacts with at point \mathbf{X}_s is randomly determined by the relative extinction coefficients (β^{air} vs. β^{aerosol}) at the voxel containing \mathbf{X}_s .

(v) If the particle is an aerosol, the photon-packet intensity is attenuated to $I_{s+1} = \varpi^{\text{aerosol}} I_s$. For a purely scattering particle, e.g. an air molecule, the photon-packet maintains its intensity. If I_{s+1} is lower than a threshold, the packet is stochastically terminated, following [23].

(vi) The photon-packet is scattered to a new random direction, determined by inverse transform sampling [24, 25], according to the phase function of the particle (Eqs. 3,4). Let $\Phi^{\text{scatter}} = \arccos(\boldsymbol{\omega} \cdot \boldsymbol{\psi})$ be the off-axis scattering angle, relative to $\boldsymbol{\psi}$. Given a random sample $u \sim \mathcal{U}[0, 1]$,

$$\Phi^{\text{scatter}} = \arccos \left\{ \frac{1}{2g} \left[1 + g^2 - \left(\frac{g^2 - 1}{1 + 2gu - g} \right) \right] \right\} \quad (11)$$

for an aerosol particles, and

$$\Phi^{\text{scatter}} = \arccos(\gamma^{\frac{1}{3}} - \gamma^{-\frac{1}{3}}) \quad \text{with} \quad \gamma = 4u - 2 + \left[(4u - 2)^2 + 1 \right]^{\frac{1}{2}} \quad (12)$$

for molecules. The scattering azimuth angle around $\boldsymbol{\psi}$ is sampled from $\mathcal{U}[0, 2\pi]$. Following this scattering event, the photon traces a new ray, denoted \mathcal{R}_{s+1} , and the next iteration of propagation (ii) proceeds.

2.1.3 Local Estimation In FMC

The quality of FMC increases with the number of photons launched. Photons contributing to any pixel are accumulated in two ways. One way is step iii above, which is a rare event. The

second way is *Local estimation* [22] which is used in conjunction to step vi, in every scattering event (Fig. 2). The local estimation contributions W_{le} expresses the probability that a photon scatters towards the camera and reaches the camera without interacting again. Let $\mathbf{V}_{s \rightarrow c}$ be the vector from the scattering point \mathbf{X}_s to \mathbf{X}_c . Let $t_{s \rightarrow c}$ be the transmittance (2) along $\mathbf{V}_{s \rightarrow c}$. Let $\Phi_{\mathcal{R}_s, c}$ be the angle between \mathcal{R}_s and $\mathbf{V}_{s \rightarrow c}$ (Fig. 2[left]). Let ϖ, P be the respective albedo and phase function of the scattering particle. If a photon scatters by an aerosol, then $\varpi = \varpi^{\text{aerosol}}$, $P = P_{\text{HG}}$. If the photon scatters by a molecule then $\varpi = \varpi^{\text{air}} \approx 1$, $P = P_{\text{Ray}}$. Local estimation then contributes

$$W_{le} = \varpi I_s P(\Phi_{\mathcal{R}_s, c}) \frac{t_{s \rightarrow c}}{|\mathbf{V}_{s \rightarrow c}|^2} \quad (13)$$

to pixel p in camera c . The factor $|\mathbf{V}_{s \rightarrow c}|^{-2}$ can be interpreted as consideration of \mathbf{X}_s to be a point radiation source. Due to this factor, FMC is unstable when the camera is *in-situ* i.e., inside the scattering medium. Local estimation from scattering points \mathbf{X}_s close to \mathbf{X}_c lead to a large increase of image variance. Hence, an infinite number of photons is needed for convergence when $|\mathbf{V}_{s \rightarrow c}| \rightarrow 0$.

2.1.4 Backward Monte Carlo Photon Tracking

Numerically, BMC is very similar to the FMC. But, there are two major difference. First, from each pixel p at camera c a photon is separately launched in direction $-\omega_p$. Then the photon is traced back through the atmosphere. A photon that happens to back-trace into the Sun, is counted as contribution to pixel p . The second difference is the local estimation calculation as we detail below. BMC take the following steps (Fig. 2):

(i) Launch a photon-packet from camera c to direction $-\omega_p$. This is the initial *ray*, denoted \mathcal{R}_0 . The packet has an initial intensity I_0 . Per iteration s :

Per iteration s :

(ii) Find the distance on ray \mathcal{R}_s to which the photon-packet propagates, as described in step (ii) of Sec. 2.1.2.

(iii) If \mathbf{X}_s is outside the domain, the packet is terminated. If $\mathcal{R}_s \parallel \omega_{\text{sun}}$, the packet is counted as contributing to pixel p .

(iv, v, vi) Sample photons scattering event: particle type, photon-packet intensity, and scattered direction, as described in steps (iv, v, vi) of Sec. 2.1.2.

Here too, local estimation is preformed in conjunction to step vi. Here local estimation derives radiance back-traced to the sun, at each scattering event. Local estimation expresses the probability that a back propagating photon scatters towards the Sun, then reaches the Sun without interacting again. Let $\mathbf{V}_{s \rightarrow \text{sun}}$ be the vector from the scattering point \mathbf{X}_s to the TOA, directed to $-\omega_{\text{sun}}$ (Fig. 2[right]). Here $t_{s \rightarrow \text{sun}}$ is the transmittance along $\mathbf{V}_{s \rightarrow \text{sun}}$, and $\Phi_{\mathcal{R}_s, \text{sun}}$ is the angle between \mathcal{R}_s and $\mathbf{V}_{s \rightarrow \text{sun}}$. Local estimation then contributes

$$W_{le} = \varpi I_s P(\Phi_{\mathcal{R}_s, \text{sun}}) t_{s \rightarrow \text{sun}}. \quad (14)$$

Since the Sun is out of the scattering medium and effectively located at infinity, there is no $|\mathbf{V}_{s \rightarrow \text{sun}}|^{-2}$ factor at all. Hence sky-images simulated by BMC are stable even in-situ. A comparison is displayed in Fig. 3: FMC rendering is very noisy compared to BMC.

3 A Proposed Forward Model

As described in Sec. 2, a BMC sky-image simulator has a major stability advantage over FMC. However BMC has drawbacks. BMC estimates radiance for one camera and one pixel at a time. In contrast, each single FMC sample trajectory can contribute to multiple viewpoints and pixels in parallel, using local-estimation (Fig. 2). This is efficient for simulating multiple cameras, which observe an atmospheric domain from N_{views} viewpoints.

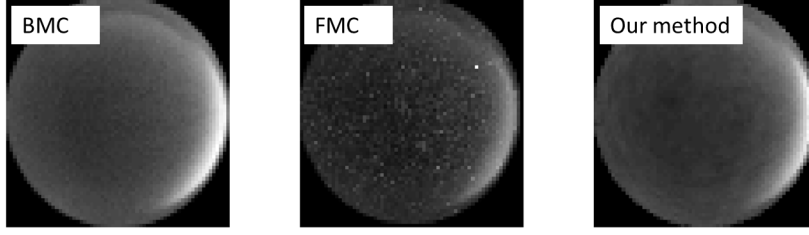


Figure 3: Comparison of rendering results. In BMC, 10^7 initial photons are used, equally divided between all pixels that view the sky. In FMC, 10^7 photons uniformly irradiate the domain TOA.

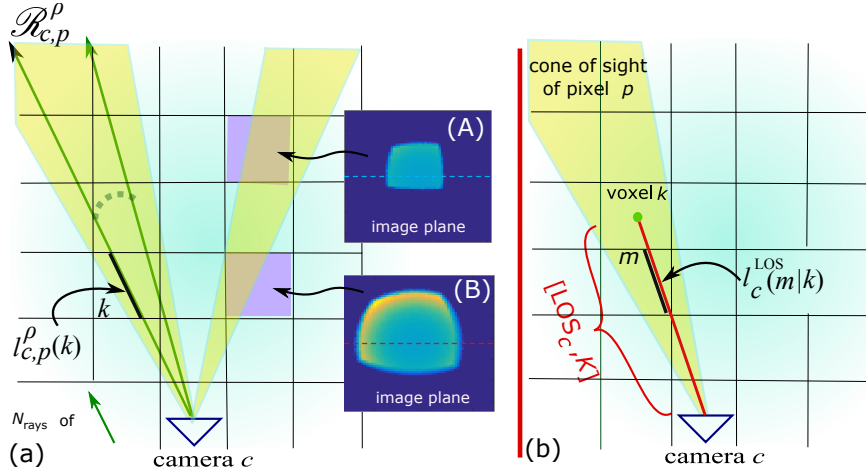


Figure 4: (a) Ray $\mathcal{R}_{c,p}^\rho$ intersects with voxel k creating a line-segment $l_{c,p}^\rho(k)$. Two isolated voxels (A) and (B) are projected to the image plane [Insets]. Because of the difference of the voxels' distance from camera c , their projected support is different. (b) Line-segment $l_c^{\text{LOS}}(m|k)$ is the intersection between voxel m and line-segment $[\text{LOS}_c, k]$.

We seek to use FMC for several reasons. First, FMC is more efficient for multi-pixel multi-view simulations. Moreover, a gradient-based recovery [19, 26], which we need in Sec. 5 requires the volumetric fields $I(\mathbf{X}, \omega)$, $J(\mathbf{X}, \omega)$, not only projected images. Volumetric fields are obtained using FMC without local estimation, hence, are not prone to instabilities. To enable FMC in-situ, however, we need to overcome the $|\mathbf{V}_{s \rightarrow c}|^{-2}$ instability. In this section, we describe a solution, disposing the $|\mathbf{V}_{s \rightarrow c}|^{-2}$ factor using voxelization of the field J .

As illustrated in Fig. 1 the volumetric domain is discretized into a grid of N_{voxels} rectangular cuboid voxels, indexed by k or m . As a numerical approximation, assume that within any voxel, the parameters $\beta(k)$, σ^{aerosol} , ϖ^{aerosol} , and g are constants, e.g., corresponding to the values at each voxel center.

Our RT solution has three steps: (i) Pre-calculate the geometry of the cameras-grid setup. (ii) FMC simulation calculates the radiance scattered from voxel k in the direction of camera c . (iii) Light attenuation along the LOS from voxel k to camera c .

3.1 Geometry

A camera sensor comprises of N_{pix} pixels. Each pixel collects light from a narrow cone in the atmosphere (Fig. 4a). The cone either contains or intersects some voxels, while remaining oblivious to the rest. The radiant power contributed by voxel k to camera c is $R_c(k)$, which we define in detail in Sec. 3.3. Overall, radiance captured at the pixel from all voxels is a weighted sum of $R_c(k)$ over all voxels k . This sum is formulated by a sparse $N_{\text{pix}} \times N_{\text{voxels}}$ matrix operation $\mathbf{\Pi}_c$, having reciprocal area units

$$\mathbf{i}_c = \mathbf{\Pi}_c \mathbf{R}_c . \quad (15)$$

Here \mathbf{i}_c is the image, column-stacked to a vector N_{pix} long, and \mathbf{R}_c is a column-stacked representation of $R_c(k)$. The weights of $\mathbf{\Pi}_c$ represent the relative portion of the radiant power contributing to camera c , solely due to geometry.

These weights are pre-calculated as follows: Divide pixel p to N_{rays} points, from each of which back-project a ray $\mathcal{R}_{c,p}^\rho$. The intersection length of $\mathcal{R}_{c,p}^\rho$ (Fig. 4a) with voxel k is $l_{c,p}^\rho(k)$. Let V_{voxel} be a voxel volume. The weight is then proportional to a normalized average intersection length,

$$\mathbf{\Pi}_c(p, k) = \bar{l}_{c,p}(k) = \frac{1}{N_{\text{rays}} V_{\text{voxel}}} \sum_{\rho=1}^{N_{\text{rays}}} l_{c,p}^\rho(k) . \quad (16)$$

Eqs. (15,16) express rendering. There is no factor proportional to $|\mathbf{V}_{s \rightarrow c}|^{-2}$ in Eqs. (15,16). This factor is implicit in the weighted sum matrix $\mathbf{\Pi}_c$: each voxel contributes to several pixels, illuminating a spot in the image plane. More rays pass through voxels closer to a camera. Thus, if a scattering event occurs in a voxel for which $|\mathbf{V}_{s \rightarrow c}|$ is small, the contribution to the image affects more pixels than if the voxel had a large $|\mathbf{V}_{s \rightarrow c}|$. This is expressed by a larger spot in the image (Fig. 4a).

3.2 Scattered Radiance Calculation with FMC

Define the *scattered radiance* as $L(\mathbf{X}, \boldsymbol{\omega}) = J(\mathbf{X}, \boldsymbol{\omega})\beta(\mathbf{X})$. Using FMC, $L(\mathbf{X}, \boldsymbol{\omega})$ can be estimated by caching all the scattering events that occurred at \mathbf{X} in direction $\boldsymbol{\omega}$. Our situation is simpler for two reasons. First, we use a voxelized radiance grid. Hence \mathbf{X} is discretized to the voxel index k . Second, as shown below, we only need to store the scattered radiance that contributes to the discrete set of N_{views} cameras $c = 1, \dots, N_{\text{views}}$. We denote the power scattered from voxel k in the direction of camera c by $L_c(k)$. For each scattering event in voxel k , update $L_c(k)$ by

$$L_c(k) \leftarrow L_c(k) + \varpi I_s P(\Phi_{\mathcal{R}_{s,c}}) . \quad (17)$$

Hence L_c is discretized in space and the relevant directions. Similarly, a discrete version of $J(\mathbf{X}, \boldsymbol{\omega})$ is

$$j_c(k) = \frac{L_c(k)}{\beta(k)} . \quad (18)$$

3.3 Optical Transmittance

The transmittance between \mathbf{X} and \mathbf{X}_c is

$$t(\mathbf{X}, \mathbf{X}_c) = \exp \left[- \int_{\mathbf{X}_c}^{\mathbf{X}} \beta(r) dr \right] . \quad (19)$$

Eq. (7) can be re-written as image rendering:

$$I(\mathbf{X}_c, \boldsymbol{\omega}_p) = A + \int_{\mathbf{X}_c}^{\mathbf{X}_{\partial\Omega}} L(\mathbf{X}', \boldsymbol{\omega}_p) t(\mathbf{X}', \mathbf{X}_c) d\mathbf{X}' , \quad (20)$$

where A represents direct solar rays entering the camera. For each camera c , denote by $[\text{LOS}_c, k]$ a LOS between camera c and the center of voxel k (Fig. 4b). Suppose this LOS intersects voxel m . The geometric length of this intersecting line segment is $l_c^{\text{LOS}}(m|k)$. Following Eq. (1), the optical depth between the center of voxel k to camera c is

$$\tau_{\text{LOS}_c}(k) = \sum_{m \in [\text{LOS}_c, k]} l_c^{\text{LOS}}(m|k) \beta(m). \quad (21)$$

Define a $N_{\text{voxels}} \times N_{\text{voxels}}$ sparse matrix whose element (k, m) is

$$\mathbf{W}_c(k, m) = \begin{cases} l_c^{\text{LOS}}(m|k) & \text{if } m \in [\text{LOS}_c, k] \\ 0 & \text{otherwise} \end{cases}. \quad (22)$$

Let $\boldsymbol{\tau}_{\text{LOS}_c}$ and $\boldsymbol{\beta}$ be column-stack vector representations of $\tau_{\text{LOS}_c}(k)$ and $\beta(k)$, respectively. Then, we can write Eq. (21) using matrix notation

$$\boldsymbol{\tau}_{\text{LOS}_c} = \mathbf{W}_c \boldsymbol{\beta}. \quad (23)$$

The discrete transmittance from the center of voxel k towards camera c is

$$T_c(k) = \exp[-\tau_{\text{LOS}_c}(k)]. \quad (24)$$

Based on Eqs. (15,17,24)

$$R_c(k) = L_c(k) T_c(k). \quad (25)$$

Let \mathbf{T}_c , \mathbf{L}_c be the column stack vector representations of $T_c(k)$ and $L_c(k)$ respectively. A column-stack vector of all voxel contributions to camera c is described by

$$\mathbf{R}_c = \mathbf{L}_c \odot \mathbf{T}_c. \quad (26)$$

Here \odot denotes the Hadamard (element-wise) product.

Let \mathbf{j}_c be a column-stack vector representation of $j_c(k)$. Then from Eqs. (15,18,26), excluding direct sun light, the image is

$$\mathbf{i}_c(\boldsymbol{\beta}) = \boldsymbol{\Pi}_c(\mathbf{j}_c \odot \mathbf{Y}_c) = \boldsymbol{\Pi}_c(\mathbf{j}_c \odot \boldsymbol{\beta} \odot \mathbf{T}_c), \quad (27)$$

where $\mathbf{Y}_c = \boldsymbol{\beta} \odot \mathbf{T}_c$. The value of pixel p in image c is

$$i_c(p) = \frac{1}{N_{\text{rays}} V_{\text{voxel}}} \sum_{\rho=1}^{N_{\text{rays}}} \sum_{k \in \mathcal{D}_{c,p}^{\rho}} l_{c,p}^{\rho}(k) j_c(k) \beta(k) T_c(k). \quad (28)$$

Eqs. (15,27,28) are a discrete version of Eq. (20), excluding the direct solar irradiance of the camera.

4 Rendering Simulations

We tested the scenes used in [16], illustrated in Fig. 5. We briefly re-mention them here for clarity.

Geometry: The atmospheric domain is $50\text{km}^2 \times 50\text{km}^2$ wide, 10km thick. The fields $\beta^{\text{aerosol}}(\mathbf{X})$ and $\beta^{\text{air}}(\mathbf{X})$ from [16] are discretized to a $20 \times 20 \times 40$ voxel grid. For rendering using the our method, the domain is more finely divided into a $80 \times 80 \times 120$ voxel grid, where the dimensions of each voxel is $625 \times 625 \times 83\text{m}$. The sun is at zenith angle $\Phi^{\text{SR}} = 45^\circ$. Following [16], the sun's red-green-blue wavelengths intensity ratios are 255 : 236 : 224. There are $N_{\text{views}} = 36$ ground-based cameras placed uniformly with $\sim 7\text{km}$ nearest-neighbor separation.

Aerosols: Two aerosols types were used all having $\varpi^{\text{aerosol}} = 1$:

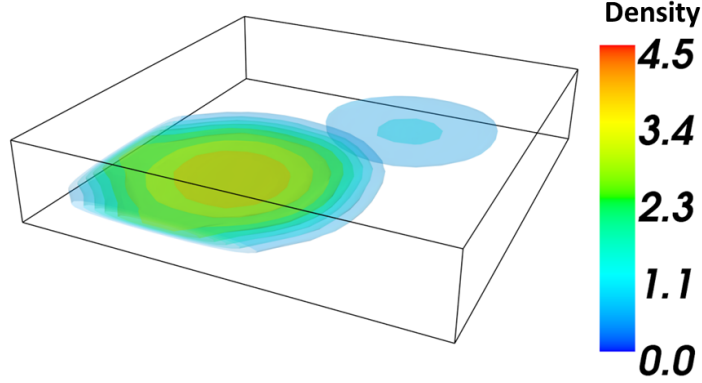


Figure 5: Aerosol distributions [16]. *Haze blobs* low density distribution. The aerosol density unit is 10^6 particles/ m^3 .

1. An artificial aerosol having an isotropic phase function. The extinction cross sections are in the red-green-blue (RGB) channels $\sigma_R^{\text{aerosol}} = \sigma_G^{\text{aerosol}} = \sigma_B^{\text{aerosol}} = 17 \mu\text{m}^{-2}$.
2. Type 6 from the aerosol list in [27]. The anisotropy parameter per color channel is $[g_R, g_G, g_B] = [0.763, 0.775, 0.786]$. The extinction cross sections are $[\sigma_R^{\text{aerosol}}, \sigma_G^{\text{aerosol}}, \sigma_B^{\text{aerosol}}] = [16.5, 16.2, 15.9] \mu\text{m}^{-2}$.

Let n^{sealevel} be a density of aerosols at sea level. Here we give a short description of the atmospheres. More details are found in [16]. We simulated different aerosol distributions:

Atm1 Haze blobs (Fig. 5) of an isotropic aerosol, at low density ($n^{\text{sealevel}} \approx 10^6$).

Atm2 Haze blobs of an anisotropic aerosol, at low density ($n^{\text{sealevel}} \approx 10^6$).

Atm3 Haze blobs of an anisotropic aerosol, at high density ($n^{\text{sealevel}} \approx 10^7$).

All the N_{views} imaging systems have a hemispherical field with $N_{\text{pix}} = 64 \times 64$.

Fig. 3 demonstrates the improvement achieved by our method relative to simple FMC (Sec. 2.1.2), on a red channel image of Atm1. Fig. 6 compares images rendered using:

1. BMC, initial 10^4 photons per pixel, i.e., $\approx 1.5 \cdot 10^9$ photons in total.
2. Proposed voxelized FMC, using initial 10^7 photons on the TOA, and $N_{\text{rays}} = 10$.
3. Single scattering approximation [16].

Rendering using our method is highly consistent with BMC rendering (Sec. 2.1.4), and similar to the single-scattering results [16]. Deviations of MC from the single-scattering results are more pronounced where multi-scatter is more significant: near the horizon and generally in Atm3. There is normalization constant between the BMC and proposed method rendered images gray level. This constant doesn't depend on atmospheric matter parameters, but on the domain geometry, and was found empirically.

Processing was preformed using MATLAB on a 2.50 GHz Intel Xeon CPU. Rendering was parallelized with 40 cores. Theoretically, For a given pre-calculated geometry (Sec. 3.1) the proposed rendering method should be $\approx 1.5 \cdot 10^9 / 10^7 = 150$ times faster. However, messaging between cores consumes additional run time, thus our method was 108 times faster than BMC.

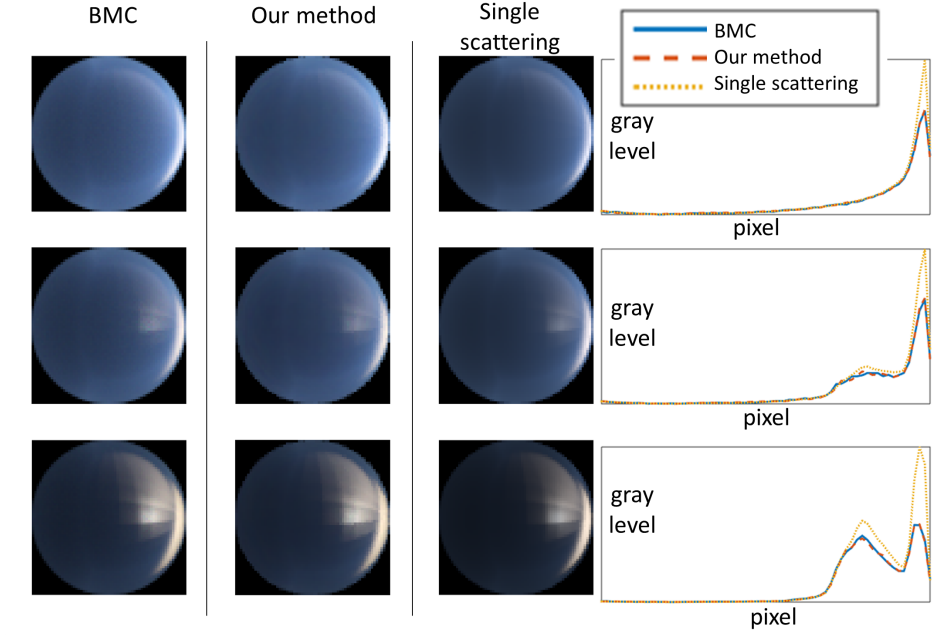


Figure 6: Image rendering and a middle horizontal line cross-section (green channel). [Top] Atm1, [middle] Atm2, and [bottom] Atm3.

5 Inverse Problem

The previous sections rendered images, assuming the field $\beta(\mathbf{X})$ is known. Now we derive and solve the inverse-problem: given acquired images, what is $\beta(\mathbf{X})$? In addition to air molecules, let there be a single type of aerosol in the atmospheric domain. Hence, the three-element vector $[\sigma^{\text{aerosol}}, \varpi^{\text{aerosol}}, g]$ is uniform across the scene and assumed known [27]. The aerosol density is spatially variable and unknown, and so is $\beta^{\text{aerosol}}(\mathbf{X}) \in \mathcal{C}$. Here \mathcal{C} comprises all possible extinction fields that comply with some constraints. Particularly, β^{aerosol} is non-negative and its spatial support is bounded between the ground and the TOA. A constraint useful for reducing the dimensionality is that β^{aerosol} is piecewise-constant, following 3D blocks of $N_x \times N_y \times N_z$ voxels. This constraint is consistent with an assumption that spatial variations of $\beta(\mathbf{X})$ are generally smooth.

The data are N_{views} image measurements $\{i_c^{\text{measured}}\}_{c=1}^{N_{\text{views}}}$. Recovery is formulated as an optimization of a cost function, to fit the image-formation model to the data [16].

$$\hat{\beta} = \arg \min_{\beta \in \mathcal{C}} \mathcal{E}(\beta) \quad (29)$$

5.1 Gradient-based Optimization

Using red-green-blue channels, the RGB extinctions are $[\beta_R, \beta_G, \beta_B]$. We solve Eq. (29) using a gradient-based method. Let the variable extinction be β_G . Let $\mu \in [R, G, B]$ be a channel color. Define $\tilde{\sigma}_\mu = \sigma_\mu^{\text{aerosol}} / \sigma_G^{\text{aerosol}}$. From the known σ^{aerosol} and β^{air} , β_G^{aerosol} determine the overall extinction per channel using

$$\beta_\mu = \beta_\mu^{\text{air}} + \tilde{\sigma}_\mu \beta_G^{\text{aerosol}}. \quad (30)$$

Let $\mathbf{i}_{c,\mu}$ and $\mathbf{i}_{c,\mu}^{\text{measured}}$ denote the modeled and measured image in channel μ , respectively. Then, the cost function of Eq. (29) is

$$\mathcal{E}(\boldsymbol{\beta}) = \sum_{\mu=R,G,B} \sum_{c=1}^{N_{\text{views}}} \left\| \mathcal{M}_c[\mathbf{i}_{c,\mu}^{\text{measured}} - \mathbf{i}_c(\boldsymbol{\beta}_\mu)] \right\|_2^2 + \eta \Psi(\boldsymbol{\beta}_G^{\text{aerosol}}). \quad (31)$$

Here Ψ is a regularization term that expresses the spatial smoothness [16] of $\boldsymbol{\beta}_G^{\text{aerosol}}$, while η is a regularization weight. In Eq. (31), \mathcal{M}_c represents masking of pixels around the Sun. There are two reasons for this masking. Sky-images estimated using MC have high variance in pixels surrounding the Sun due to the stochastic nature of MC. The problem occurs as the phase function is sampled at forward scattering angles.² A sun-mask avoids use of noisy modeled pixel. Moreover, a sun occluder is typically applied cameras, to block lens flare and saturation [10].

Let $(\cdot)^\top$ denote transposition. Then, the gradient of Eq. (29) is

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{\beta}} = 2 \sum_{\mu=R,G,B} \tilde{\sigma}_\mu \sum_{c=1}^{N_{\text{views}}} \mathbf{Q}_c \left\{ \mathcal{M}_c \mathcal{J}_c(\boldsymbol{\beta}_\mu) \right\}^\top \mathcal{M}_c [\mathbf{i}_c(\boldsymbol{\beta}_\mu) - \mathbf{i}_{c,\mu}^{\text{measured}}] + \eta \frac{\partial}{\partial \boldsymbol{\beta}_G^{\text{aerosol}}} \Psi(\boldsymbol{\beta}_G^{\text{aerosol}}). \quad (32)$$

Here the matrix $\mathcal{J}_c(\boldsymbol{\beta}_\mu)$ is the Jacobian of the vector $\mathbf{i}_{c,\mu}$ with respect to $\boldsymbol{\beta}_\mu$, and \mathbf{Q}_c is a diagonal weighting matrix which is detailed in Sec. 6. Element (p, k) of $\mathcal{J}_c(\boldsymbol{\beta}_\mu)$ differentiates the intensity of pixel p in viewpoint c with respect to the extinction at voxel k ,

$$\mathcal{J}_c(\boldsymbol{\beta}_\mu) = \begin{bmatrix} \partial i_{c,\mu}(1)/\partial \beta_\mu(1) & \dots & \dots & \dots & \partial i_{c,\mu}(1)/\partial \beta_\mu(N_{\text{voxels}}) \\ \partial i_{c,\mu}(2)/\partial \beta_\mu(1) & \dots & \dots & \dots & \partial i_{c,\mu}(2)/\partial \beta_\mu(N_{\text{voxels}}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \partial i_{c,\mu}(p)/\partial \beta_\mu(1) & \vdots & \partial i_{c,\mu}(p)/\partial \beta_\mu(k) & \dots & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ \partial i_{c,\mu}(N_{\text{pix}})/\partial \beta_\mu(1) & \dots & \dots & \dots & \partial i_{c,\mu}(N_{\text{pix}})/\partial \beta_\mu(N_{\text{voxels}}) \end{bmatrix}. \quad (33)$$

Estimating Eq. (33) using Eq. (27), is very complex. The reason is that Eqs. (6,20) express a recursive interplay of the fields I, J , that are functions of $\boldsymbol{\beta}$ (see Eqs. 5,7,20). It is complex to perform recursion per each gradient component.

In [16, 26], a similar forward model is applied while recovery is done by least squares minimization. The forward model in [16] assumes single scattering for image rendering. Under the single scattering assumption, the unidirectional Sun rays scatter at most once, thus \mathbf{j}_c is easily calculated: Eq. (6) degenerates to an integral over a δ function of orientation. Then, the Jacobian $\mathcal{J}_c(\boldsymbol{\beta}_\mu)$ has a closed-form expression. Using MC for image rendering, there is no close-form solution for the gradient (32). The forward model in [26] uses FMC to render images of a homogeneous medium. Minimization in [26] is solved by a stochastic gradient descent, where the unknown parameters are the spatially uniform $\{\beta, \varpi, P(\cdot)\}$. Gradient computation in [26] is intensive, with a set of three cascade FMC simulations per optimization iteration. Trying a similar formulation in a heterogeneous medium would mean $\mathcal{O}(N_{\text{voxels}})$ FMC renderings per iteration, since there are N_{voxels} degrees of freedom. This approach is computationally very expensive on large grids.

²To reduce noise in forward scattering, some works [23, 29] approximate the phase function or use *importance sampling*.

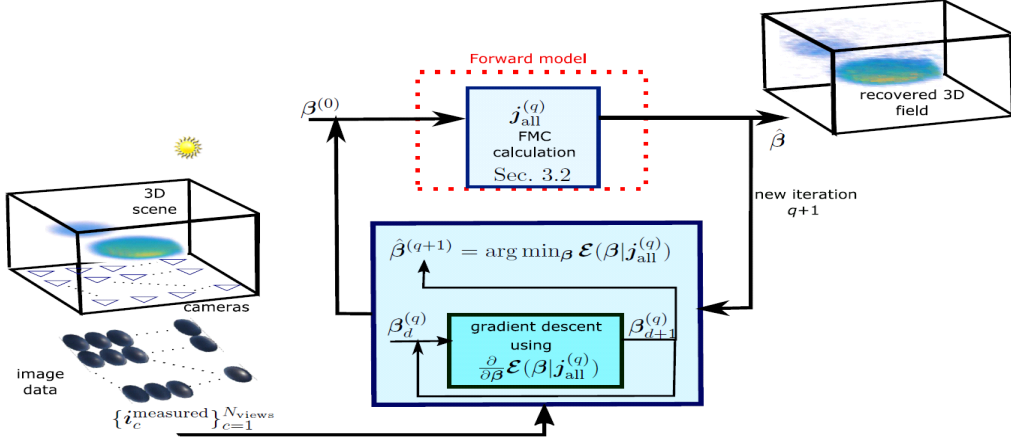


Figure 7: Iterative optimization process using surrogate function $\mathcal{E}(\beta | j_{\text{all}}^{(q)})$.

5.2 An Efficient Approach

Instead of a direct estimation of Eqs. (32,33) we optimize β using a *surrogate function* [19]. Denote by $j_{c,\mu}$ a field j_c in channel μ . Let $j_{\text{all}} = \{\{j_{c,\mu}\}_{c=1}^{N_{\text{views}}}\}_{\mu=R,G,B}$. For a fixed j_{all} , Eq. (29) is easily minimized as we explain. Gradient-based optimization is iterative. Define $\beta^{(q)}$ as an estimation of β in the q 'th iteration. Based on $\beta^{(q)}$, the field $j_{\text{all}}^{(q)}$ is computed using FMC (Sec. 3.2). This step is *not* an inverse problem but forward-model rendering. Consequently, the computational complexity of this step does not increase with N_{voxels} .

After $j_{\text{all}}^{(q)}$ is derived, it is fixed for a while. Denote $\mathcal{E}(\beta | j_{\text{all}}^{(q)})$ as a surrogate function for a fixed $j_{\text{all}}^{(q)}$. Keeping $j_{\text{all}}^{(q)}$ fixed, β is evolved. The following iterative optimization process is defined

$$\hat{\beta}^{(q+1)} = \arg \min_{\beta \in \mathcal{C}} \mathcal{E}(\beta | j_{\text{all}}^{(q)}). \quad (34)$$

Fig. 7 summarizes the iterative optimization process.

As we now show, when $j_{\text{all}}^{(q)}$ is fixed, the Jacobian (33) degenerates to a simple calculation. We now detail the derivation of the Jacobian $\mathcal{J}_c(\beta | j_{c,\mu}^{(q)})$ for a given fixed array $j_{\text{all}}^{(q)}$, i.e., $\partial j_{c,\mu}^{(q)} / \partial \beta_\mu \equiv 0$. Let $T_{c,\mu}$ and $Y_{c,\mu}$ be the fields T_c and Y_c in channel μ , respectively. Let $\mathbb{D}\{v\}$ denote conversion of a general vector v into a diagonal matrix, whose main diagonal elements correspond to the elements of v . Using Eqs. (23,24,27) and expressions from [16] for the gradient of element-wise products, the Jacobian (33) degenerates to

$$\mathcal{J}_c(\beta_\mu | j_{c,\mu}^{(q)}) = \frac{\partial \Pi_c(j_{c,\mu}^{(q)} \odot Y_{c,\mu})}{\partial \beta_\mu} = \Pi_c \mathbb{D}\{j_{c,\mu}^{(q)}\} \frac{\partial Y_{c,\mu}}{\partial \beta_\mu}, \quad (35)$$

$$\frac{\partial Y_{c,\mu}}{\partial \beta_\mu} = \mathbb{D}\{\beta_\mu\} \frac{\partial T_{c,\mu}}{\partial \beta_\mu} + \mathbb{D}\{T_{c,\mu}\}, \quad (36)$$

and

$$\frac{\partial T_{c,\mu}}{\partial \beta_\mu} = -\mathbb{D}\{\exp[-W_c \beta_\mu]\} W_c. \quad (37)$$

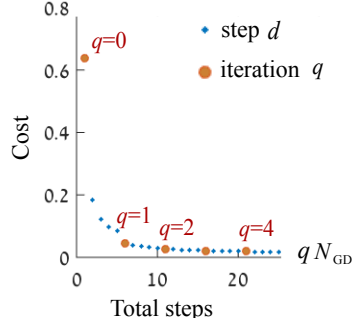


Figure 8: Example of cost minimization using $N_{\text{GD}} = 5$.

Given $\mathcal{J}_c(\beta_\mu | \mathbf{j}_{c,\mu}^{(q)})$, Eqs. (34,32) can be solved using *Gradient Descent*

$$\beta_{d+1}^{(q)} = \beta_d^{(q)} - \Delta \frac{\partial}{\partial \beta} \mathcal{E}(\beta | \mathbf{j}_{\text{all}}^{(q)}), \quad (38)$$

where d indexes a gradient descent step, and Δ is the step size. After every N_{GD} gradient descent steps, \mathbf{j}_{all} is updated. Then, gradient-descent of β is resumed, using the updated \mathbf{j}_{all} for another set of N_{GD} gradient descents, and so on until convergence. Fig. 8 shows an example of the cost minimization using $N_{\text{GD}} = 5$. A major advantage of the surrogate function is effective gradient calculation, without N_{voxels} rendering processes.

6 Conditioning the Optimization

Figure 9a illustrates a test atmosphere observed by 25 ground based cameras. In Fig. 9b, maximum intensity projections (MIP) [28] visualizes $\beta_1^{(0)}$. This is the very first step ($d = 1$) in the first iteration ($q = 0$). The field $\beta_1^{(0)}$ in Fig. 9b stems from $\mathbf{j}_{\text{all}}^{(0)}$, which was created by the initialization $\beta^{\text{aerosol}} = 0$. The field contains artifacts. The artifacts appear as high values of $\beta_1^{(0)}$ at voxels near the in-situ cameras. These voxels are unstable. This problem is comparable to ill-conditioned linear optimization.

If all cameras are far from the scattering domain, all voxels are similarly observed [19], and there is no conditioning problem. However, in-situ, voxels affect unequally the data (Fig. 4a). Due to geometry, if voxel k projected to more pixels than voxel m then, $\partial \mathcal{E} / \partial \beta(k)$ tend to be significantly higher than $\partial \mathcal{E} / \partial \beta(m)$. Depending on Δ , this imbalance leads to instability in nearby voxels or very slow convergence. This problem is solved by conditioning, achieved using a diagonal weighting matrix \mathbf{Q}_c . Element (k, k) of \mathbf{Q}_c is the number of rays $\mathcal{R}_{c,p}^p$ that pass through voxel k . The field $\beta_1^{(0)}$ that evolves using this conditioning is visualized in Fig. 9c. This field has much weaker artifacts.

In addition, Eq. (31) includes a regularization term. Our regularization term [16] is $\Psi(\beta_{\text{G}}^{\text{aerosol}}) = \|\mathbf{W} \mathcal{L} \beta_{\text{G}}^{\text{aerosol}}\|_2^2$, where \mathcal{L} is a matrix representation of the 3D Laplacian operator. The matrix \mathbf{W} is diagonal: its elements are a function of the altitude of each voxel [16]. The gradient of $\Psi(\beta_{\text{G}}^{\text{aerosol}})$ which used in Eq. (32), is $2\mathcal{L}^T \mathbf{W}^T \mathbf{W} \mathcal{L} \beta_{\text{G}}^{\text{aerosol}}$.

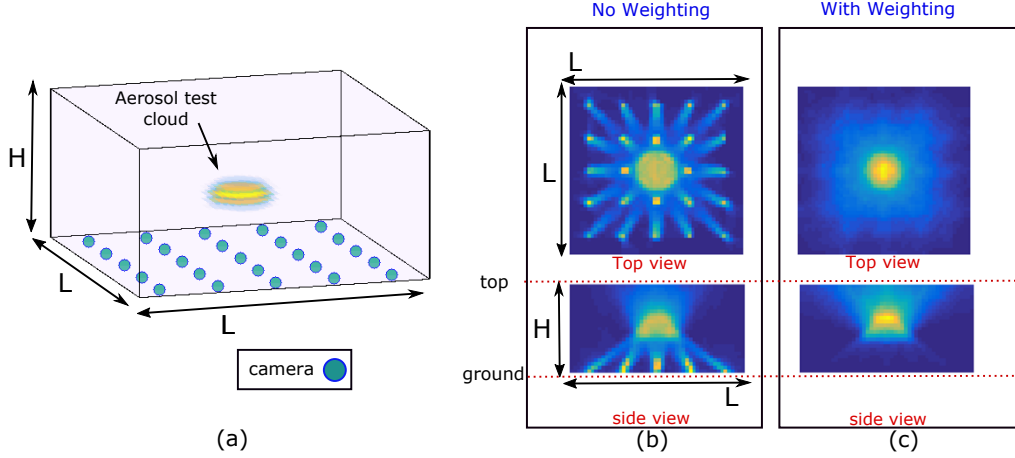


Figure 9: (a) The domain is observed by 25 ground-based cameras. An elliptic aerosol cloud at the center of an air-filled domain. (b,c) Top view and side view MIP visualizations of $\beta_1^{(0)}$.

7 Recovery Simulations

7.1 Images and Noise

For the scenes described in Sec. 4, a set $\{i_c^{\text{measured}}\}_{c=1}^{N_{\text{views}}}$ was rendered using BMC (Sec. 2.1.4) since BMC is the most accurate and precise method, despite its slow speed. These images are noisy because MC sampling implicitly induces Poissonian noise. This naturally mimics photon noise in optical imaging. To fully simulate a camera, we incorporate scaling of optical energy to graylevels in a 10-bit camera, read noise and quantization. These operations are expressed by

$$i_c^{\text{measured}} \leftarrow i_c^{\text{measured}} \left(\frac{2^{10}}{\max\{\{\mathcal{M}_c i_c^{\text{measured}}\}_{c=1}^{N_{\text{views}}}\}} \right) + o_{\text{noise}}. \quad (39)$$

The read-noise by o_{noise} is white, with standard deviation of 0.4 graylevels. The values in Eq. (39) are clipped to the range $[0 \dots 1024]$ and rounded. The resulting images are the input for our reconstruction method and its comparison to [16].

7.2 Recovery Results

In addition to the three aerosol distributions in Sec. 4, the following distribution is used as in [16]:

Atm4 Haze Front of an anisotropic aerosol, at low density, shown in Fig. 10 ($n^{\text{sealevel}} \approx 10^6$).

The Haze Front is rendered as an elliptic cylinder intersected by the domain, as detailed in [16].

The optimization was initialized by $\beta^{\text{aerosol}} = 0$, or by a results [16] obtained using single scattering approximation. The analysis used the following parameters: $\Delta = 10^{-3}$ and $N_{\text{GD}} = 5$. Starting from $\beta^{\text{aerosol}} = 0$, satisfactory convergence occurred after several hundred iterations.

The total estimation error [16] is quantified by the aerosol mass that is over and under-estimated in all voxels, relative to the total aerosol mass in the scene. Using the ℓ_1 norm, the total mass relative difference is $\delta_{\text{mass}} = (\|\hat{\mathbf{n}}\|_1 - \|\mathbf{n}^{\text{true}}\|_1) / \|\mathbf{n}^{\text{true}}\|_1$. In order to sense local errors [16], we use $\epsilon = \|\hat{\mathbf{n}} - \mathbf{n}^{\text{true}}\|_1 / \|\mathbf{n}^{\text{true}}\|_1$. Some reconstructions are illustrated in Fig. 10.

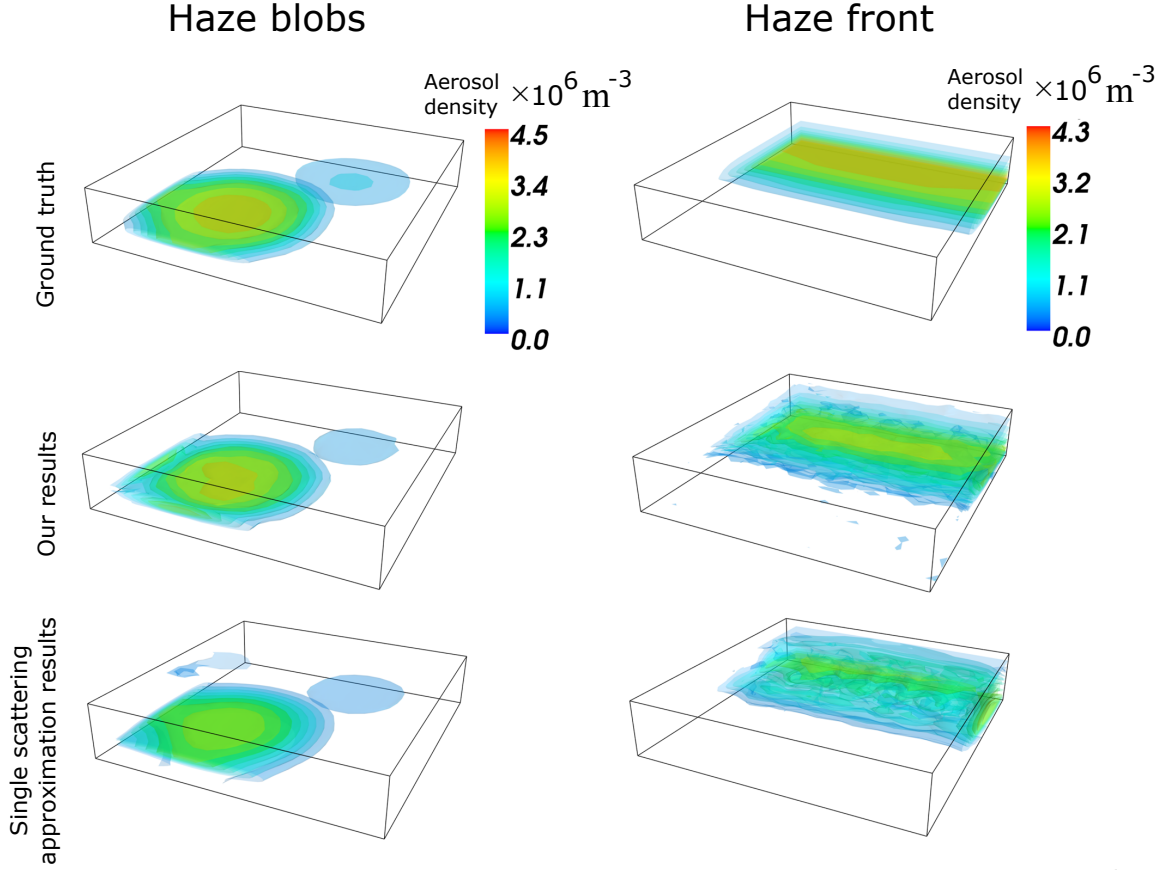


Figure 10: [Top] Ground-truth aerosol distributions: Atm1 (Haze blobs, [left]), and Atm4 (Haze front, [right]). Color encodes aerosol density [10^6 particles/ m^3]. [Middle] Our recontraction . [bottom] Reconstruction using the single scattering approximation [16].

Scene	Single scattering		Our method null initialization		Our method, initialized by single scattering	
	δ_{mass}	ϵ	δ_{mass}	ϵ	δ_{mass}	ϵ
Atm1	1.7%	50%	3.4%	26%	3%	23%
Atm2	-6.3%	61%	10%	38%	11%	37%
Atm3	14%	63%	4.1%	27%	7%	28%
Atm4	23%	76%	2.4%	70.8%	5.7%	43%

Table 1: Comparison of relative errors of our recovery method to single scattering approximation method from [16].

Additionally, Table 1 summaries and compares the quantitative results for all the described distributions. In conclusion, our results are better than [16]. Thus, accounting for multiple

scattering is important and significant, while being computationally feasible.

Acknowledgements:: We are grateful to Anthony Davis, Raanan Fattal, Michael Zibulevsky for fruitful discussions. We thank Mark Sheinin, Johanan Erez, Ina Talmon, Dani Yagodin for support. YYS is a Landau Fellow - supported by the Taub Foundation. His work is conducted in the Ollendorff Minerva Center. Minerva is funded through the BMBF. AL acknowledges ISF and ERC for financial support.

References

- [1] R. Fattal. Single image dehazing. In *Proc. ACM TOG*, 27:72, 2008.
- [2] C. Fuchs, M. Heinz, M. Levoy, H.-P. Seidel, and H. Lensch. Combining confocal imaging and descattering. In *Proc. EGSR*, 27:1245–1253, 2008.
- [3] E. Namer, S. Shwartz and Y. Y. Schechner Skyless polarimetric calibration and visibility enhancement, *Optics express* 17:472-493, 2009.
- [4] S. G. Narasimhan, S. K. Nayar, B. Sun, and S. J. Koppal. Structured light in scattering media. In *Proc. IEEE ICCV*, 420–427, 2005.
- [5] Y. Y. Schechner, D. J. Diner and J. V. Martonchik, Spaceborne underwater imaging, In *Proc. Proc. IEEE ICCP*, 2011.
- [6] T. Treibitz and Y. Y. Schechner, Polarization: Beneficial for visibility enhancement? In *Proc. IEEE CVPR*, 2009.
- [7] Y. Tian and S. G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *Proc. IEEE ICCV*, 2303–2310, 2009.
- [8] N. J. Morris and K. N. Kutulakos. Dynamic refraction stereo. In *Proc. IEEE ICCV*, 1573–1580, 2005.
- [9] X. Zhu and P. Milanfar. Stabilizing and deblurring atmospheric turbulence. In *Proc. IEEE ICCP*, 2011.
- [10] D. Veikherman, A. Aides, Y. Y. Schechner, and A. Levis. Clouds in the cloud. *Proc. ACCV*, 2014.
- [11] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel. Time-resolved 3D capture of non-stationary gas flows. In *Proc. ACM TOG*, 27:132, 2008.
- [12] C. Ma, X. Lin, J. Suo, Q. Dai, and G. Wetzstein. Transparent object reconstruction via coded transport of intensity. In *Proc. IEEE CVPR*, 3238–3245, 2014.
- [13] B. Trifonov, D. Bradley, and W. Heidrich. Tomographic reconstruction of transparent objects. *Proc. EGSR*, 2006.
- [14] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar. Refractive shape from light field distortion. In *Proc. IEEE ICCV*, 1180–1186, 2011.
- [15] P. Modregger, M. Kagias, S. Peter, M. Abis, V. A. Guzenko, C. David, and M. Stampanoni. Multiple scattering tomography. *Phys. Rev. Lett.*, 113:020801, 2014.
- [16] A. Aides, Y. Y. Schechner, V. Holodovsky, M. J. Garay, and A. B. Davis. Multi sky-view 3D aerosol distribution recovery. *Opt. Express*, 21:25820–25833, 2013.
- [17] D. A. Boas, D. H. Brooks, E. L. Miller, C. A. DiMarzio, M. Kilmer, R. J. Gaudette, and Q. Zg. Imaging the body with diffuse optical tomography. *IEEE Signal Proc. Mag.*, 18(6):57–75, 2001.
- [18] A. Gibson, J. Hebden, and S. R. Arridge. Recent advances in diffuse optical imaging. *Phys. Med. Biol.*, 50(4):R1, 2005.

- [19] A. Levis, Y. Y. Schechner, A. Aides and A. B. Davis. Airborne Three-Dimensional Cloud Tomography. In *Proc. IEEE ICCV* 2005.
- [20] S. Chandrasekhar. *Radiative Transfer*. Dover Pub., 1960.
- [21] L. Devroye. Sample-based non-uniform random variate generation *Proceedings of the 18th conference on Winter simulation*, (ACM, 1986), pp. 260–26.5
- [22] A. Marshak and A. B. Davis. *3D Radiative Transfer in Cloudy Atmospheres*. Springer, 2005.
- [23] H. Iwabuchi. Efficient Monte Carlo Methods for Radiative Transfer Modeling *J. Atmos. Sci.*, 63:2324–2339, 2006
- [24] J. R. Frisvad. Importance sampling the Rayleigh phase function *JOSA.*, 26(12):2436–2441, 2011
- [25] T. Binzoni, T. S. Leung, A. H. Gandjbakhche, D. Rüfenacht, D. T. Delpy. The use of the Henyey–Greenstein phase function in Monte Carlo simulations in biomedical optics *Phys. Med. Biol.*, 51(17):313–322, 2006
- [26] I. Gkioulekas, S. Zhao, K. Bala, T. Zickler, and A. Levin. Inverse volume rendering with material dictionaries. *Proc. ACM TOG*, 32:162, 2013.
- [27] J. V. Martonchik, R. A. Kahn, and D. J. Diner, “Retrieval of aerosol properties over land using MISR observations,” in *Satellite Aerosol Remote Sensing over Land*, , A. A. Kokhanovsky and G. Leeuw, eds. (Springer Berlin Heidelberg, 2009), pp. 267–293.
- [28] J. W. Wallis, T. R. Miller, C. A. Lerner, and E. C. Kleerup. Three-Dimensional Display in Nuclear Medicine In *Proc. IEEE T-MI*, 8(4), 1989
- [29] R. Buras ,B. Mayer. Efficient unbiased variance reduction techniques for Monte Carlo simulations of radiative transfer in cloudy atmospheres: The solution *J. Quant. Spectrosc. Radiat. Transfer*, 112:434–447, 2011
- [30] Y. Y. Schechner, S. K. Nayar, P. N. Belhumeur. Multiplexing for optimal lighting In *Proc. IEEE TPAMI*, 29(8):1339–1354, 2007